

5 DAT 元件

5.1 介绍

DAT 元件主要用来处理数据。他们可以编辑、分析、创建、收发多种类型的数据。比如文本字符串、列表、Python 脚本、XML、JSON、MIDI、串口、OSC。

项目中的逻辑处理，主要就靠 DAT 元件和 Python 脚本。比如分析列表数据信息，管理其他元件状态、通过外部通讯执行复杂任务，等等。后面会有一些这方面的例子。

TD 可以看作是一个模块化的 Python 编程环境。复杂、长篇的 Python 工程和函数，在这里被拆分，变成轻量化、模块化的片段。就像 TextDAT 元件那样。这种设定易于学习、维护、和扩展的特性，大大方便了分享和项目协作。

5.2 通讯方式

TD 原生支持 MIDI、OSC、TCP、UDP、UDT 以及 Websocket，所以它能跟多种系统如 Web 应用和服务、其他计算机系统、显示控制硬件等通讯。

CHOP 元件中的 MIDI、OSC、DMX 协议在 DAT 元件中也适用。

TCP 是因特网的标准通信协议。它是一个面向连接的协议，在通讯双方中有明确的客户端和服务端，并且在传输数据前，先要进行握手连接。这种连接是可靠的，通讯双方可以校验确认所有数据都已经被接收，也就是没有数据丢失。TCP 是一种顺序流通讯，发送方按一定顺序发送，接收方也按一定顺序接收。

而 UDP 是一种无连接的协议。数据发送前，不需要建立连接。UDP 是不可靠的，可能会产生丢包、粘包等问题，不过它的通讯速度更快。

UDT 是一种新的通讯协议，它是 TCP+UDP 的优化版。它基于 UDP，但是是有连接的可靠通讯。它用了 TCP 那套校验规则，还保持 UDP 的收发速度。

无连接协议有个好处是能多播。比如你的系统里有一大堆电脑，多播能一次性发给网络中的全部电脑。避免了逐个发给每台电脑产生的性能开销。相对应的单播，就得一台一台发。

Websocket 用于跟 Web 浏览器和实时 Web 应用通讯。它在一些双向网络通讯的功能上做了简化。